

Diskussionen über Latenz, Durchsatz und Vectorized Hash Tables

Latenz und Durchsatz

Die Metriken Latenz und Durchsatz geben Aufschluss über die Leistung einer intrinsischen Funktion:

Latenz bezieht sich auf die Anzahl der Zyklen, die eine intrinsische Funktion benötigt, bis ihr Ergebnis verfügbar ist und für weitere Berechnungen verwendet werden kann. Eine niedrigere Latenz bedeutet, dass das Ergebnis einer Operation schneller verfügbar ist, was für die Gesamtleistung kritisch sein kann, insbesondere in Szenarien, in denen jede Operation auf dem Ergebnis der vorherigen aufbaut.

Durchsatz gibt an, wie viele Zyklen vergehen, bis die nächste gleiche Art von intrinsischer Operation gestartet werden kann. Ein höherer Durchsatz bedeutet, dass mehr Operationen desselben Typs in einer gegebenen Zeit gestartet werden können, was die parallele Ausführung von Berechnungen und die Nutzung der CPU-Ressourcen verbessert.

Zusammen geben diese Metriken ein umfassendes Bild davon, wie gut eine intrinsische Funktion unter verschiedenen Bedingungen performt. Die Latenz ist besonders relevant für die Ausführungsgeschwindigkeit einzelner Operationen, während der Durchsatz zeigt, wie gut die Funktion in einem parallelen Verarbeitungskontext skaliert. Eine optimale Nutzung intrinsischer Funktionen erfordert oft einen Ausgleich zwischen diesen beiden Metriken, abhängig von den spezifischen Anforderungen der Anwendung und der Arbeitslast.

Intrinsische Funktionen, die eine geringe Latenz und einen hohen Durchsatz bieten, sind besonders wertvoll in Leistungs-kritischen Anwendungen, wie zum Beispiel in der digitalen Signalverarbeitung, Grafikberechnungen und wissenschaftlichen Simulationen, wo große Datenmengen effizient verarbeitet werden müssen. Die Wahl und Anwendung der richtigen intrinsischen Funktionen können somit maßgeblich zur Leistungssteigerung von Software beitragen.

Vectorized Hash Tables

Vectorized Fingerprinting (VFP)

- VFP is an extension that directly vectorizes linear probing, but it is limited to integer keys and has a low degree of parallelism dependent on the size of the key. It also requires access to three arrays. To address these issues, prior work utilized Bucket-Based Comparison (BBC) and introduced VFP as an intermediate step to separate the reasons for potential performance gains. VFP operates similarly to linear probing but uses fingerprints, i.e., 8-bit or 16-bit key hashes, and resolves conflicts only in case of a match. VFP allows operations to be independent of the key data type, as everything is either an 8-bit or 16-bit integer fingerprint. It can compare more information at once due to the smaller size of fingerprints compared to 32-bit or 64-bit keys, eliminates the need for an additional validity information array by defining a certain fingerprint as invalid (e.g., 0), and stores keys and values in the same array.

Bucket-Based Comparison (BBC)

- BBC represents the hash table as an array of buckets, each holding fingerprints, metadata, and key-value pairs. BBC has three main advantages over VFP: firstly, it avoids the additional latency of obtaining the key-value pair since fingerprints and data are co-located; secondly, it stores a flag indicating whether a bucket has overflowed, providing an opportunity for early termination; thirdly, BBC separates the interfaces of the hash table and buckets, simplifying and clarifying the implementation. Based on the key's fingerprint and bucket index, it checks whether the bucket contains the key by comparing the fingerprints and then the keys upon finding a match. If the bucket has overflowed, the process continues with the next bucket. BBC avoids unaligned loads, increases data locality, and makes the implementation more compact.

These methodologies demonstrate how to implement efficient vectorized hash schemes on modern CPU architectures, aimed at enhancing the performance and efficiency of data processing systems when handling large volumes of data.